Back to Deep Learning Nanodegree

Generate Faces

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congrats!

This is one of the best projects that I have reviewed. The network is perfectly working and generating images.
I encourage you to spend time going through these to further enhance your knowledge:

- http://3dgan.csail.mit.edu
- https://medium.freecodecamp.org/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394
- https://arxiv.org/abs/1406.2661
- https://cs.stanford.edu/~ermon/papers/imitation_nips2016_main.pdf

Required Files and Tests

The project submission contains the project notebook, called "dlnd_face_generation.ipynb".

The project submission contains all the files required for reviewing the project.

All the unit tests in project have passed.

All the unit tests are running perfectly.

Build the Neural Network

The function model_inputs is implemented correctly.

The tensorflow placeholders of real_inputs, random noise z and learning rate are rightly created. Their shapes, data types and ranks are correctly declared.

The function discriminator is implemented correctly.

Amazing!

The discriminator network is perfectly created.
The usage of convolution layers, batch normalization and leaky relu are precisely done.

The function generator is implemented correctly.

Perfect!

The architecture of the generator network is perfect. Deconvolution layers are precisely used to generate the images. Thumbs up for adding 2 additional layers to increase the depth of the network.

Suggestion

It would be great if you use kernel initializer like Xavier initializer to initialize the weights.

The function model_loss is implemented correctly.

The model_loss() function is correctly coded to compute the generator-discriminator losses.

The function model_opt is implemented correctly.

Neural Network Training

The function train is implemented correctly.

- It should build the model using `model_inputs`, `model_loss`, and `model_opt`.
- It should show output of the `generator` using the `show_generator_output` function

The train() function is correctly building the model by using model_inputs, model_loss, and model_opt functions.

Required

```
_ = sess.run (g_opt, feed_dict={input_z: batch_z, input_real: batch_images, lr: lea
rning_rate})
```

I noticed that you have ran the generator optimizer two times. Didn't get the reason of using it 2 times? Please consider rectifying this.

The parameters are set reasonable numbers.

```
batch_size = 32
z_dim = 100
learning_rate = 0.001
beta1 = 0.4
```

```
batch_size = 64
z_dim = 100
learning_rate = 0.001
beta1 = 0.5
```

The hyperparameter values are well tuned.

The project generates realistic faces. It should be obvious that images generated look like faces.

Awesome!

The generated images are perfect.

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ